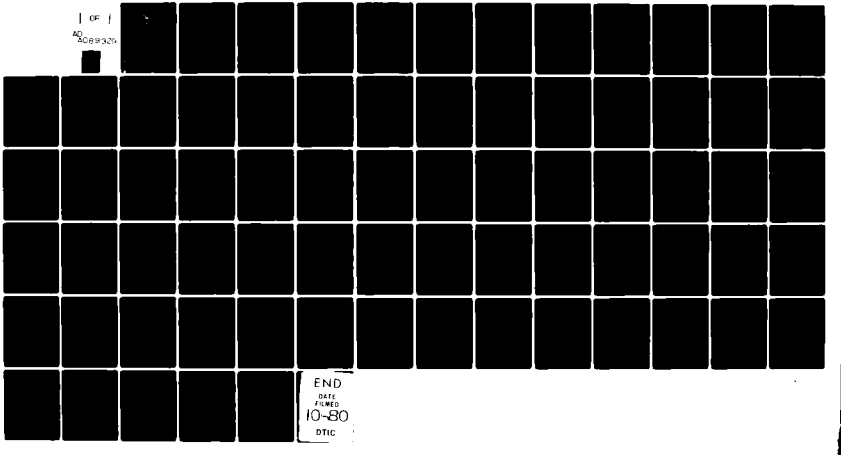
AD-A089 329 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL--ETC F/G 9/2
SOFTWARE CONTROL DURING DEVELOPMENT AND ACQUISITION.(U)
JUN 80 R J LAMKEY, C T PAVY
UNCLASSIFIED AFIT-LSSR-62-80 NL



LEVEL I

AD A089329

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY (ATC)

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

SOFTWARE CONTROL DURING DEVELOPMENT AND ACQUISITION

Robert J. Lamkey, Captain, USAF
Curtis T. Pavy, GS-11

LSSR 62-80

The contents of the document are technically accurate, and no sensitive items, detrimental ideas, or deleterious information are contained therein. Furthermore, the views expressed in the document are those of the author(s) and do not necessarily reflect the views of the School of Systems and Logistics, the Air University, the Air Training Command, the United States Air Force, or the Department of Defense.

USAF SCN 75-20B AFIT Control Number LSSR 62-80

AFIT RESEARCH ASSESSMENT

The purpose of this questionnaire is to determine the potential for current and future applications of AFIT thesis research. Please return completed questionnaires to: AFIT/ LSH (Thesis Feedback), Wright-Patterson AFB, Ohio 45433.

1. Did this research contribute to a current Air Force project?

   a. Yes b. No

2. Do you believe this research topic is significant enough that it would have been researched (or contracted) by your organization or another agency if AFIT had not researched it?

   a. Yes b. No

3. The benefits of AFIT research can often be expressed by the equivalent value that your agency received by virtue of AFIT performing the research. Can you estimate what this research would have cost if it had been accomplished under contract or if it had been done in-house in terms of man-power and/or dollars?

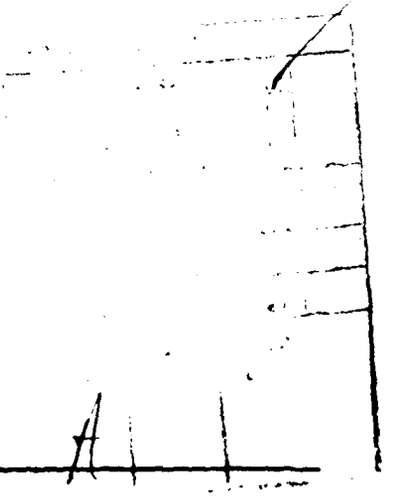
   a. Man-years _____ $ _____ (Contract).

   b. Man-years _____ $ _____ (In-house).

4. Often it is not possible to attach equivalent dollar values to research, although the results of the research may, in fact, be important. Whether or not you were able to establish an equivalent value for this research (3 above), what is your estimate of its significance?

   a. Highly Significant b. Significant c. Slightly Significant d. Of No Significance

5. Comments:



Name and Grade _____ Position _____

Organization _____ Location _____

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE, $300

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 73236 WASHINGTON D.C.

POSTAGE WILL BE PAID BY ADDRESSEE

AFIT/LSH (Thesis Feedback)
Wright-Patterson AFB OH 45433

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: AFIT-LSSR-62-80 (14)	2. GOVT ACCESSION NO.: AD-A089329
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): SOFTWARE CONTROL DURING DEVELOPMENT AND ACQUISITION. (6)	5. TYPE OF REPORT & PERIOD COVERED: Master's Thesis (9)
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Robert J. Lamkey, Captain, USAF (10) Curtis T. Pavy, GS-11	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Graduate Education Division School of Systems and Logistics Air Force Institute of Technology, WPAFB OH	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS (11)
11. CONTROLLING OFFICE NAME AND ADDRESS: Department of Communication and Humanities AFIT/LSH, WPAFB OH 45433	12. REPORT DATE: Jun 80 (12) 73 13. NUMBER OF PAGES: 60
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

APPROVED FOR PUBLIC RELEASE AFR 190-17.

Fredric C. Lynch

FREDRIC C. LYNCH, Major, USAF
Director of Public Affairs

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Software
Control of Development Processes
Software Control
Software Development
Acquisition Procedures

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Thesis Chairman: Steven Lathrop, Captain, USAF

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

012250

The Air Force has experienced some difficulty in obtaining quality software under specified cost, schedule, performance criteria. This thesis was undertaken to explore the underlying problem and to research methods for improving Air Force software acquisitions. The literature highlighted a number of problem areas evident in Air Force and DOD in general. The major problem areas were: (1) lack of measurable milestones, (2) lack of consideration for the integration of hardware and software, (3) lack of software visibility during development, and (4) lack of user involvement. The researchers used those problem areas as a basis for conducting interviews at a major non-DOD software user/producer. The effort identified aspects of a software control process with potential applications for Air Force use. The researchers discovered that an effective software control process is feasible, but it rests upon a realization of the unique nature of software. Also, the impact of quality assurance, user involvement, and development planning upon the final software product are discussed.

LSSR 62-80

SOFTWARE CONTROL DURING DEVELOPMENT AND ACQUISITION

A Thesis

Presented to the Faculty of the School of Systems and Logistics

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Logistics Management

By

Robert J. Lamkey, BA
Captain, USAF

Curtis T. Pavy, BS
GS-11

June 1980

This thesis, written by

Captain Robert J. Lamkey

and

Mr. Curtis T. Pavy

has been accepted by the undersigned on behalf of the Faculty of the School of Systems and Logistics in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN LOGISTICS MANAGEMENT
(ACQUISITION LOGISTICS MAJOR)

DATE: 9 June 1980

COMMITTEE CHAIRMAN

ACKNOWLEDGMENTS

We would like to gratefully acknowledge the individuals at NCR Corporation who assisted us in this research effort. Without their help and cooperation this research could not have been accomplished. Further, we wish to thank our thesis advisor, Captain Steven Lathrop, for his advice and assistance and our typist, Cecilia Weaver, for her patience, help and high quality of workmanship.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Background

The first electronic digital computer was built under the direction of Harvard professor Howard Aiken, working with IBM engineers. It was completed in 1944 and utilized electromagnetic relays along with mechanical counters to automatically execute arithmetic and logic operations. The only human intervention required was the initial wiring for the job to be accomplished and starting the machine.

Due to the demands of World War II, the United States Army funded the development of ENIAC (Electronic Numerical Integrator and Calculator). The primary reason for this computer was to compute and compile ballistics tables, therefore, it did not have general purpose capabilities. This marked the entry of the Department of Defense (DOD) into the computer field.

In 1946, a mathematician named John Von Newmann, in collaboration with two others, suggested that instructions for computers, as well as data, could be stored internally. This could be accomplished through the use of a binary numbering system. The concept of internally stored computer instructions was incorporated into the EDVAC (Electronic Discreet Variable Automatic Computer). EDVAC was completed

in 1952 and was the beginning of software as it is known today.

The development of semiconductors and integrated circuits in the 1960s started the movement of the computer industry toward small minicomputers. Also, in the early 1970s the development of micro-processor chips further reduced the physical size of computers while increasing the computational power and the ability to store and process information. These developments have brought about the use of computers in such things as the ignition systems of automobiles and opened up the new field of embedded computers[1] (2:60-67).

Just as computers are varied in ability and application, there are many types of computer languages, each with its own merits and particular application advantages. Also, there are varying design techniques available to the programmer which may enhance the usefulness of software[2] once it has been developed. All of these factors complicate the management of software acquisition and development.

---

[1]An embedded computer is defined as a computer which operates within, and as part of, a large piece of equipment or hardware system.

[2]For the purposes of this thesis, software is defined as the programs and/or instructions to the hardware which are not "hardwired" into the system and can be changed through non-hardware modifications.

There are at least 450 general-purpose programming languages and dialects used in DOD computer applications. Each is incompatible with the others, and none are widely used (9:26). The only possible exceptions are COBOL and FORTRAN, because there is a standard for each language. However, the standard form is seldom used. Since so many of these languages are incompatible, transfer of information becomes a problem and complicates the management of a total system which may contain several computers and program languages.

Today, computers are used to store, process, and calculate information in everything from simple children's toys to very sophisticated missile guidance systems. The past twenty-five years has seen the advent of "striking increases in computing speed, memory capacity, and hardware reliability, with simultaneous decreases in power consumption and hardware cost [9:24]." Concurrently, increased demands for sophisticated and specialized software have facilitated a dramatic increase in the proportion of total computer system costs attributable to software. As shown in Figure 1, software comprised less than 20 percent of total system costs in 1955; today over 80 percent of computer system costs are for software.

The DOD has experienced numerous problems during the development and acquisition of software. Many studies, by

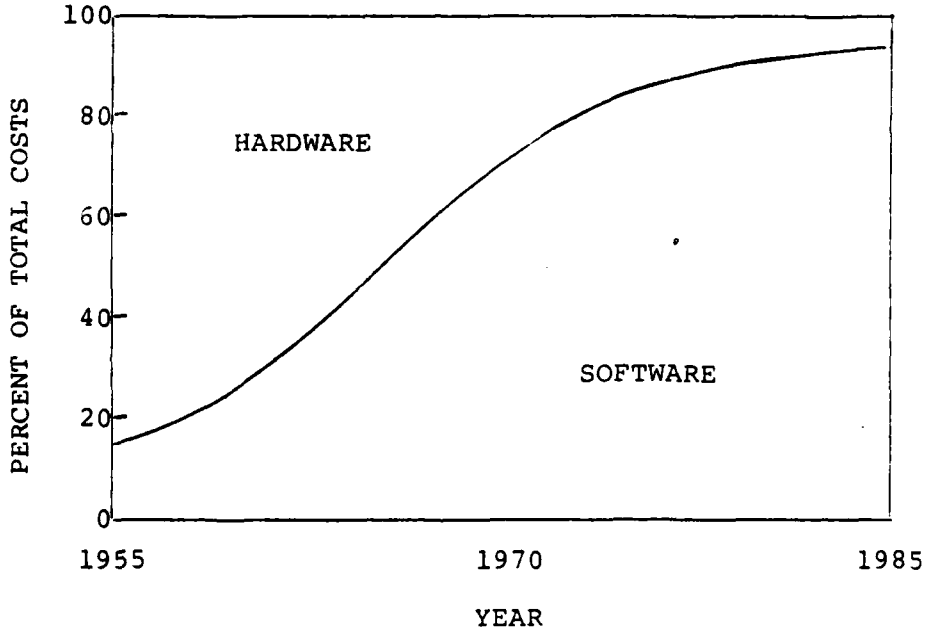


Fig. 1. Hardware/software cost trends (1:16)

government researchers and consultants, have been conducted to explore and recommend solutions to the identified problems. One of the most frequently recognized problem areas is the lack of control[3] of software development by the system program office (SPO) during the acquisition process. The lack of control is a consequence of deficiencies in current software acquisition regulations and the prevailing attitude of managers concerning the nature of software (10:2-4).

A host of directives exist that apply, in one way or another, to the development and acquisition of computers and software. Figure 2 depicts the major software acquisition directives; the points of the arrows indicate their range of application within the acquisition process. As indicated, some of the directives are pertinent throughout the acquisition process while others apply over portions of the process. Little functional guidance is given as to managing software acquisitions, and software control is not specifically addressed (11).

In the past, system program managers have tended to treat software as data. In addition, they allowed software acquisition to be buried in major system acquisitions.

---

[3]Control is defined as the ability to influence and manage a process in order to meet milestones, cost targets, and other requirements.

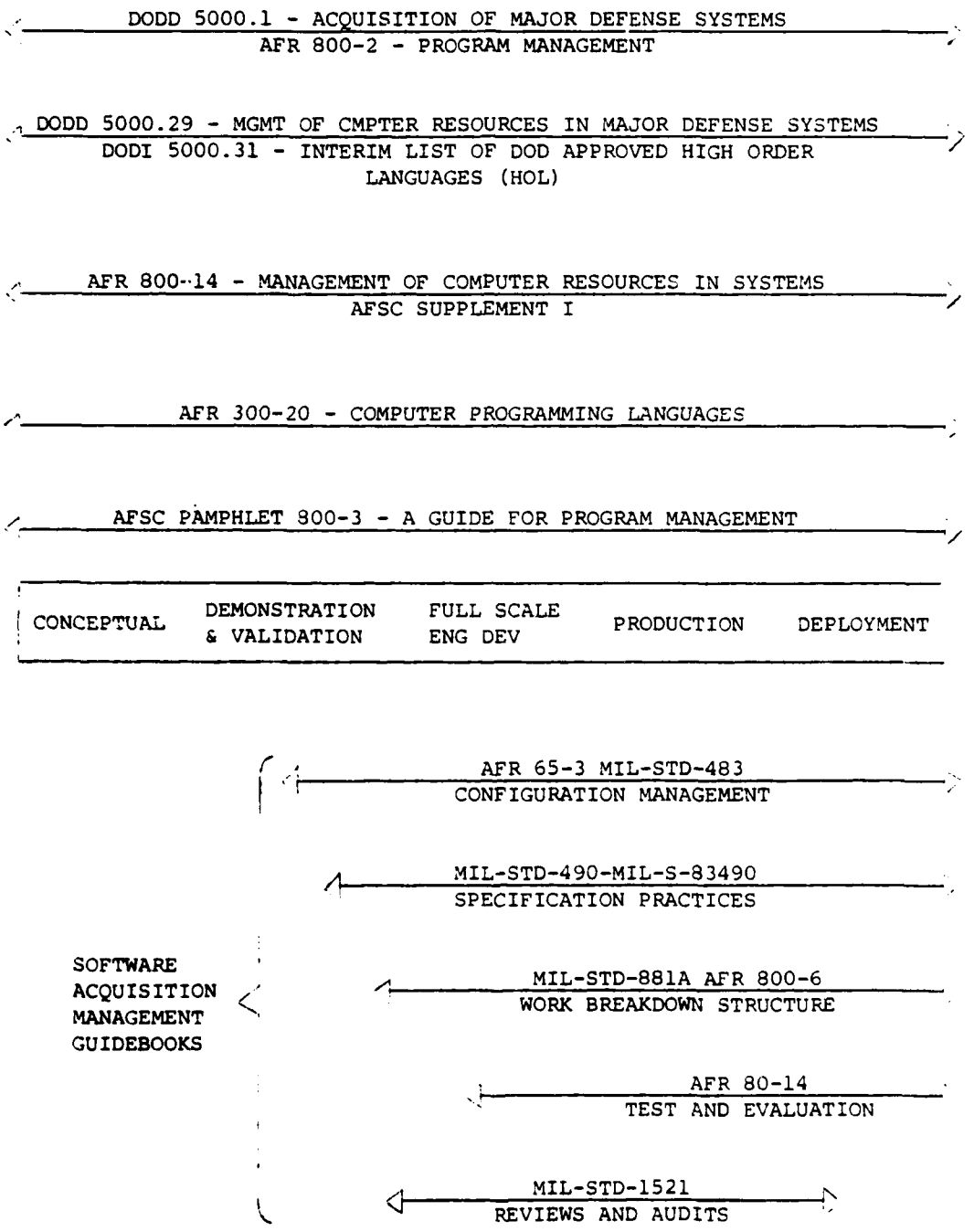


Fig. 2. Software Acquisition Management Directives (10:34)

Those practices led to a lack of visibility during the development stage of the acquisition process (10:2) and were contributory to cost overruns and degraded system performance. Due to recent comments by General Alton Slay, Commander of the Air Force Systems Command, displaying an interest in the management methods and policies of private industry, it was determined that a study of a major civilian company would be beneficial (14). In an attempt to keep the government regulations and procedures from influencing the management and control processes, a non-defense related company was selected. Therefore, the thrust of this thesis will be to identify and examine how a major civilian user of software deals with factors impacting upon the software control problem.

Problem Statement

Software within systems required by the Air Force is critical since the systems cannot function properly without it. However, management control of software in the acquisition and development process is not adequate, resulting in frequent cost overruns, schedule slippages, inadequate performance, and an inability to use the system as originally envisioned. It is the intent of this research to study the acquisition and development of software in private industry. From this study, techniques may be discovered which could

enhance the Air Force's ability to control the acquisition and development of software.

Literature Review

In order to gain a more comprehensive understanding of factors bearing on the problem and characteristics involved in the control of software acquisition and development, a literature review was conducted. This review was necessary for two basic reasons: (1) to fully understand the scope of the problem in order to evaluate the main areas of research, and (2) to determine that research in these areas would not be duplicated by this research effort.

The literature identified a number of factors contributing to the poor control of software. The most fundamental problems stem from a lack of appreciation of the uniqueness of software and its importance to the total computer system. Regulations which govern the acquisition process of software do not adequately consider its unique nature. While there exists a large number of Air Force regulations dealing with software acquisition, they appear to be modifications of hardware and item type related regulations. Since the wording and language used in the regulations address equipment, this indicates a lack of management attention towards software (5:10).

Alan J. Driscoll cited three methods of improving control of software development:

> Get the user involved early. Require an early statement of user requirements and meaningful participation in design reviews.
>
> Insist on full incorporation of software into the system requirement analysis process. Software must be engineered as an integral part of the weapon system.
>
> Place software at a high level in the WBS and remove it from the category of "data" [6:25].

Winston W. Royce also emphasizes the need for user involvement. He states: ". . . The user of the software must be capable of injecting his expertise into the software product . . . [13:1-21]."

Getting the user involved in the development process enhances the definition of requirements, specifications, and minimizes errors in estimates of resource requirements, productivity, and reliability. Communication is believed to be an essential ingredient in the assurance of software quality, and should be maintained to the highest degree possible (12).

Another factor cited as contributory towards inability to control costs and performance is the lack of measurable milestones. Software is currently managed under a system of phases coupled with reviews. However, review criteria are often not formally stated and the reviews tend to become "quite variable and subjective [12:34]." Cost control during software development would be greatly enhanced by having the ability to detect deviations or problems as early as possible. A study on software cost estimation methods for Electronic Systems Division stated:

". . . an orderly development process with measurable milestones [7:10]" is essential for effective software control.

A frequently encountered flaw in software control during development was the lack of integration of hardware and software. Hardware and software were designed or changed without giving proper attention to the other component. This often resulted in costly and inefficient operating systems. The need for close integration of software and hardware is paramount (6).

The lack of consideration for the uniqueness of software during systems development is illustrated by placing it at a low level in the Work Breakdown Structure[4]. This has caused a lack of visibility, and therefore, very little management attention to be given to the development of software. Also, this has contributed to cost overruns and system degradation.

In summary, the most significant factors cited in the literature which contribute to poor software control in DOD are:

---

[4]The Work Breakdown Structure (WBS) is a contractualy mandated arrangement by which a system acquisition is factored into subsystems and subcomponents of scheduled work packages or major tasks. The purpose of WBS is to define tasks for ease of measurement and evaluation of a contractor by the government.

1. Lack of effective or qualitative user involvement.

2. Lack of measurable or quantitative milestones.

3. Lack of consideration of hardware and software integration.

4. Lack of software visibility during systems development.

Although other factors bearing on the software control problem were identified in the literature, they were basically related to the four previously stated.

Scope and Objectives

There is no substantial production phase in software. Once software has been developed, tested, and determined to be adequate, it takes only a matter of minutes to copy and verify the copy of the software which has been developed. With today's technology there is no problem in copying software from one disk to another or from one tape to another. If necessary, cards or even listings may be used to transfer software from one system to another system. Even the maintenance of existing software could be looked at as the development of a new "part" of the program which can be "fit" into the existing one. Therefore, the primary area of control needs to be in the development portion of the life-cycle of software.

Since there are requirements that contractors fit their organization to the Work Breakdown Structure, they are almost forced to follow the same management control procedures as the government. To control out government procedures, policies, and regulations it was decided that a non-defense related corporation should be compared to the Air Force in the management and control area of software development and acquisition. Since NCR Corporation is the second largest corporation in the computer and data processing industry (15:1111), and since they purchase, as well as develop, the software which they sell, NCR Corporation was picked as a non-defense related company from which the Air Force may possibly learn. Their geographic nearness to the Air Force Institute of Technology and willingness to participate in the study were important considerations in light of practical cost and time constraints placed on the research effort.

A descriptive study appeared to be the most appropriate and productive approach to research. Essentially, because the research will not test hypotheses but will focus upon detailing or describing aspects of NCR Corporation's software control system, the vehicle for research will be the research question. Through knowledge about the Air Force's management techniques gained from the literature review, a comparison can be made. Policies and management techniques used by NCR will be evaluated to determine if

any benefits to the Air Force could be obtained by the utilization of these management techniques. Also, recommendations and areas of further research will be identified.

In an attempt to limit and control the scope, four major elements were determined to be the most critical, and this research will be restricted to these elements. First, the appreciation of NCR for the unique nature of software will be determined through the amount of visibility maintained. The amount of visibility will be evaluated by determining whether or not software development is controlled separately from hardware, but with an equal level of importance. Second, the amount of user involvement will be evaluated. Therefore, whether or not the user is part of the development effort throughout the process will be determined. Third, it is believed that hardware and software integration should be controlled. For this reason, the amount of software control, and how it relates to the integration between hardware and software will be evaluated. Fourth, since management requires identification of problem areas and progress measurement, the existence and use of milestones or deadlines must be considered.

From this, five research questions were developed. This research effort will answer those questions as they pertain to NCR. The questions are as follows:

1. Does NCR control software development and/or acquisition as a separate element?

2. Does NCR involve users throughout the development and/or acquisition process?

3. Does NCR closely integrate the acquisition and/or development of software with hardware?

4. Does NCR use milestones in the development and/or acquisition process, and if so, how?

5. What effect do the policies identified by the questions above have on the performance, cost, and schedule of software development at NCR?

The fifth question was added in order to judge the degree of influence the answers to the first four questions have on the success of a software development effort.

Justification of Research

> Software is finding its way into the critical path of many more defense systems. Performance critical software can be found in all extremes--from the large World-Wide Military Command and Control System (WWMCCS) and Trident Fleet Missile (FBM) down to miniaturized flight control packages for missiles and aircraft [4:310].

The DOD is currently spending in excess of $3 billion per year on software for embedded defense computer systems (4:309). Therefore, software costs will continue to be, as previously noted, a significant factor in computer system costs. It is imperative, because of the cost factor, that

software acquisition policies be employed which will insure software quality and conservation of public funds. However, existing policies have been lacking, resulting in ". . . large cost overruns, schedule slippages, inadequate performance, and an inability to use the system as originally envisioned [3:1]."

One aspect of the software acquisition process identified as problematic is the lack of established policies to control software during the development phase. If the problem of inadequate control of software during the development phase is to be rectified, a pertinent and useable set of management policies must be established. An exploration into the software development control policies of private industry may provide insight leading to the establishment of pertinent and useable software control policies by the Air Force.

CHAPTER II

RESEARCH METHODOLOGY

Overview

In the first chapter a problem was identified: management control of software by the Air Force during the development phase of the acquisition process is inadequate. An investigation of the literature indicated that the control of software was deficient primarily because software was not considered as a separate element during system acquisition, user involvement during development was not adequate, software and hardware integration was generally poor, and a specified system of measurable milestones for software acquisitions did not exist. It was decided that this work would examine how a major non-DOD affiliated corporation controls software, using the previously identified causes of poor software control in the Air Force as a guide. Five research questions were developed to ascertain how NCR, the selected test corporation, deals with control of software, and what impact the policies identified have on the quality of the product in terms of performance, cost and schedule. Those five questions were:

1. Does NCR control software development and/or acquisition as a separate element?

2. Does NCR involve users throughout the development and/or acquisition process?

3. Does NCR closely integrate the acquisition and/or development of software with hardware?

4. Does NCR use milestones in the development and/or acquisition process, and if so, how?

5. What effect do the policies identified by the questions above have on the performance, cost, and schedule of software at NCR?

Instrument Selection

A host of factors impacted upon the choice of research method and design. However, the answers to the newspaperman's questions of "who, what, when, where, why" provided basic but excellent guidance to the "how" or research design. Based on preliminary discussions with NCR, the questions and answers were formulated as follows:

Q. Who has the needed data?

A. Managers at NCR involved in software development and acquisition.

Q. What types of data will be needed?

A. Much of the data will be qualitative (opinions, management techniques, policies): some may be quantitative (cost, figures, performance data).

Q. Where are the respondents located?

A. Respondents are proximally located and easily accessible to the researchers.

Q. When may the respondents be reached?

A. Times must be scheduled by mutual agreement.

Q. Why was NCR chosen?

A. NCR is a major user and supplier of software and has expressed an enthusiastic willingness to participate in a study of software control. In addition, since both NCR and the Air Force organically develop and contract for the development of software, similar management problems are expected to be encountered. After considering the intent of the thesis, the level and nature of data desired, and the attitude and availability of NCR, a personal interview format (the how) was decided upon. Emory (8:Ch.10) attributed the ability to probe, control, elaborate and adjust as prime benefits of a personal interview. However, he also cautioned that care must be taken to prevent interviewer actions from biasing responses.

In order to reduce impact of interviewer bias and to minimize the effects of the inexperience of the interviewers, a carefully constructed set of interview questions was developed. The series of questions, used as an interview guide, was constructed with the awareness that sequencing, wording, sensitivities of the respondent, and content

of questions are as important as a conducive interview environment and an unbiased question delivery. The questionnaire was designed to present simple demographic "ice breaker" questions, then gradually proceed to more complex questions. Questions which are associated or logically related were grouped together to provide as much continuity as possible. With the rationale behind the choice of method and design established, an examination of what data were needed prompted the development of the interview questions. Due to the flexibility and control the interview method offers, and since it allows follow-up questions in the event an answer to any given question is inadequate, other methods, such as a telephone or mail survey, were regarded as inadequate.

Interview Question Development

In order to organize and structure the interviews, a set of interview questions was developed. The basic research questions were evaluated as to where they fit in a general acquisition process and factors impacting the questions were identified with respect to that phase of the process. Then, the interview questions were designed around these factors. In some cases, it was found that a single factor could have a relationship, or bearing on more than one question. For example, the use of milestones could be an indication of high software visibility, and

provide information about the level of hardware and software integration.

Separate Control of Software Development

Software should be treated with a high degree of individual attention and consideration because of its unique nature. So, by no means should it be treated as merely data. As previously mentioned, separate controls over the software and hardware development processes should provide the required amount of visibility, and therefore, enhance the end product. For this reason, whether or not NCR has separate control over the software development will be determined.

User Involvement

Indications of potential users' involvement are many. The first involvement would normally come during the requirements determination phase. While it is logical that potential users of a product would be allowed input during the requirements determination phase of product development, the amount of input and the extent of involvement could have a large bearing on the cost, useability, and success of the product in question. Therefore, the degree to which potential users are involved in the review process of software development is also an area which must be considered.

Integration of Hardware and Software Development and Acquisition

Software and hardware must work together. A computer, by itself, is only a piece of machinery and wiring. It is its ability to accept software and do a multitude of different types of jobs that make the computer such a powerful tool. Since each computer must be able to read, interpret, and understand the software which is fed into it, the software type and design is dependent upon the hardware in which it must function. On the other hand, the computer can only accomplish what it has been instructed to do by the software given to it. Therefore, the amount of integration between software and hardware during the development phase of a new product is important.

Use of Milestones

Milestones can be an effective tool to compare actual performance against the expected performance. This is illustrated by the wide use of milestones in the acquisition process, the contracting process, planning, training, and many other areas. If milestones are used by NCR during the development of software, the extent of their usage must be determined. The point at which milestones are set in the development process, whether or not cost estimates are matched with the milestones, how milestones are used to evaluate performance of the software developers, and how

the milestones and performance evaluations are utilized in the progress reviews were all questioned.

The Instrument

An extensive list of questions was developed with consideration given to each of the factors mentioned above. The questions were then grouped together by basic area of concern. During the grouping process, duplicate questions were eliminated. After further analysis, five major categories and points of particular interest in each category were identified. The questions were then reviewed and updated to reflect this categorization. This made up the basic framework of the interview questions. Routine questions such as name, position, and experience were then added. What follows are the interview questions which will be used as a guide to conduct the interviews with NCR management personnel.

1. Name.
2. Position in NCR.
3. How many years experience with NCR?
4. How many years experience in your field?
5. How are product requirements determined?
    a. What decisions are made whether or not to meet the requirements?
    b. What type of justification is required?

c. Are potential users consulted during this process?

6. How are cost estimates made for product development?
   a. How are these estimates updated?
   b. At what point in the process do you consider cost estimates?
   c. Are user inputs considered?
7. What type of general standards are used to determine performance levels of product development within the company?
   a. How are they determined for development outside the company?
   b. What type of reviews are made?
   c. What type of financial controls are utilized?
      (1) How are they updated?
      (2) How effective is this process?
   d. Are milestones used in the development process?
      (1) Are they formal or informal?
      (2) How are the milestones updated?
      (3) How effective is this process?

8. What type of review process is used during development?
    a. At what organizational levels are they conducted?
    b. How often are they conducted?
    c. Are potential users involved in the review?
    d. What financial considerations are made in this review process?
    e. How are decisions made to continue or discontinue development?
9. How are software and hardware trade-offs determined?
    a. At what organizational level are they made?
    b. Where, in the development process, are they made?
    c. What factors are considered?
10. Is there anything you would like to add?
11. Are there any questions I can answer for you?

The Subjects

The final aspect of the research methodology was the selection of subjects. Prior to selecting subjects, however, it was necessary to understand the organizational structure of NCR and the function of software and software related departments. It was imperative that the researchers have this information to intelligently determine which

departments held the answers to the research questions. An overview of NCR's organizational structure, and information on where best to seek answers to the research questions was provided during a day-long preliminary interview with a senior software manager. The departments of interest were selected by the researchers subsequent to the preliminary interview. The preliminary interview also provided an opportunity to test and validate the interview questions. During the course of the interview, each of the research questions was addressed to get both a greater understanding of NCR's organizational structure and an idea of the usefulness of the interview guide. The researchers were very satisfied that the instrument was indeed valid and useful; the responses to the questions provided the type of information needed to answer the research questions.

The departments selected ranged from the lowest functional level of software management to very high policy making echelons of software management. In order not to reveal potentially company classified information, specific organizational departments will not be discussed. The researchers wanted a spectrum of management levels to better understand the extent and perceive usefulness of NCR's software control policies. The actual subjects were selected with the advice of the aforementioned senior software manager. It was he who arranged the schedule of interviews

and acted as a liaison between the researchers and the subjects. The subjects, seven in all, ranged in computer experience from eleven to twenty-three years, and were employed with NCR between two and twenty-four years.

CHAPTER III

ANALYSIS AND FINDINGS

Overview

In this chapter, several items will be discussed. First, some comments about the research opportunity will be made. Second, the method used to analyze the responses obtained during the interviews will be outlined. Third, a brief description of the control system utilized will be presented. Fourth, responses to each of the research questions will be discussed. Fifth, some of the perceived benefits of NCR's control method will be mentioned. Since the control procedure employed by NCR is company classified, the details of their system will not be mentioned.

Comments

The procedure used at NCR to control software development was relatively new, not much more than one year old. Even though the procedure is updated, modified, and improved with time, it provided an excellent opportunity for research at this time. The managers at NCR had time to use the new procedure, and the first revision had just been published when this research effort was conducted. Also, managers interviewed were able to make

comparisons between the old and new procedures since they had used each.

Seven managers at NCR were interviewed. The level of management responsibility of the individuals interviewed ranged from immediately below vice-presidential through first-line supervisors and management staff personnel. Also, the respondents represented quality assurance, production management, software development, and corporate software management areas.

It is these areas which are involved in the control of development, development, and quality of the software at NCR.

Analysis

During the interviews, separate notes were taken by each author. This was done for two reasons. First, since both individuals took notes, the chance that information would be missed was decreased. Second, this technique provided a control mechanism against personal bias by the individual taking the notes. After the interviews had been conducted, the notes were compared. Through this process of comparison, an attempt was made to minimize bias.

After the comparison of the notes had been completed, an analysis of the system used to control software was done. The authors had been briefed by NCR on the new procedure

which is used to control software development. Therefore, each of the responses received during the interviews could be evaluated with respect to their procedure. This evaluation was done not only with respect to the control procedure employed by NCR, but also with respect to the five research questions. The responses from each of the managers indicated that they all operated in agreement with NCR's control procedure. Also, with respect to the research questions, responses from each of the managers interviewed were in agreement with each other. The authors realize that there is seldom this much agreement among individuals. However, since the purpose of this research effort was to discover and evaluate the systems and procedures employed by NCR, this agreement strengthens the fact that a "standard" method of control is used. Further, since the underlying subject of the interview questions was the control procedures, a high degree of correlation among the managers is not surprising.

The Control Process

Prior to the implementation of the new software control procedure, NCR had used a method of control which was patterned after a hardware development control system. The managers at NCR's headquarters determined that this type of control procedure did not work. Therefore, another method of controlling software development had to be found. The

new method which was implemented is a modification of Philip Metzger's which is described in his text, Managing a Programming Project (12). The authors found this fact interesting since, as stated in Chapter I, the Air Force's regulations seem to be adaptations of hardware control procedure, and the Air Force is currently having difficulty controlling their software acquisitions.

Since the details of the procedure used at NCR is company classified, they will not be discussed here. However, a brief description will be given as it relates to Mr. Metzger's method. This method of control is illustrated in Figure 3.

A great deal of attention is given to each software development project during the definition and design phases. As illustrated in the figure, it is at this time that the specifications which the software must meet are defined. Also, the method of design and the testing procedures are specified.

During the planning activity, milestone dates and cost targets are set. As the project progresses through the cycle, the milestones and cost targets are monitored by management to identify problem areas as early as possible. Through early identification of problem areas, corrective action can be taken in order to meet the schedules. Also, a project may be broken down into modules which may or may not be developed separately from one another. This, too, is

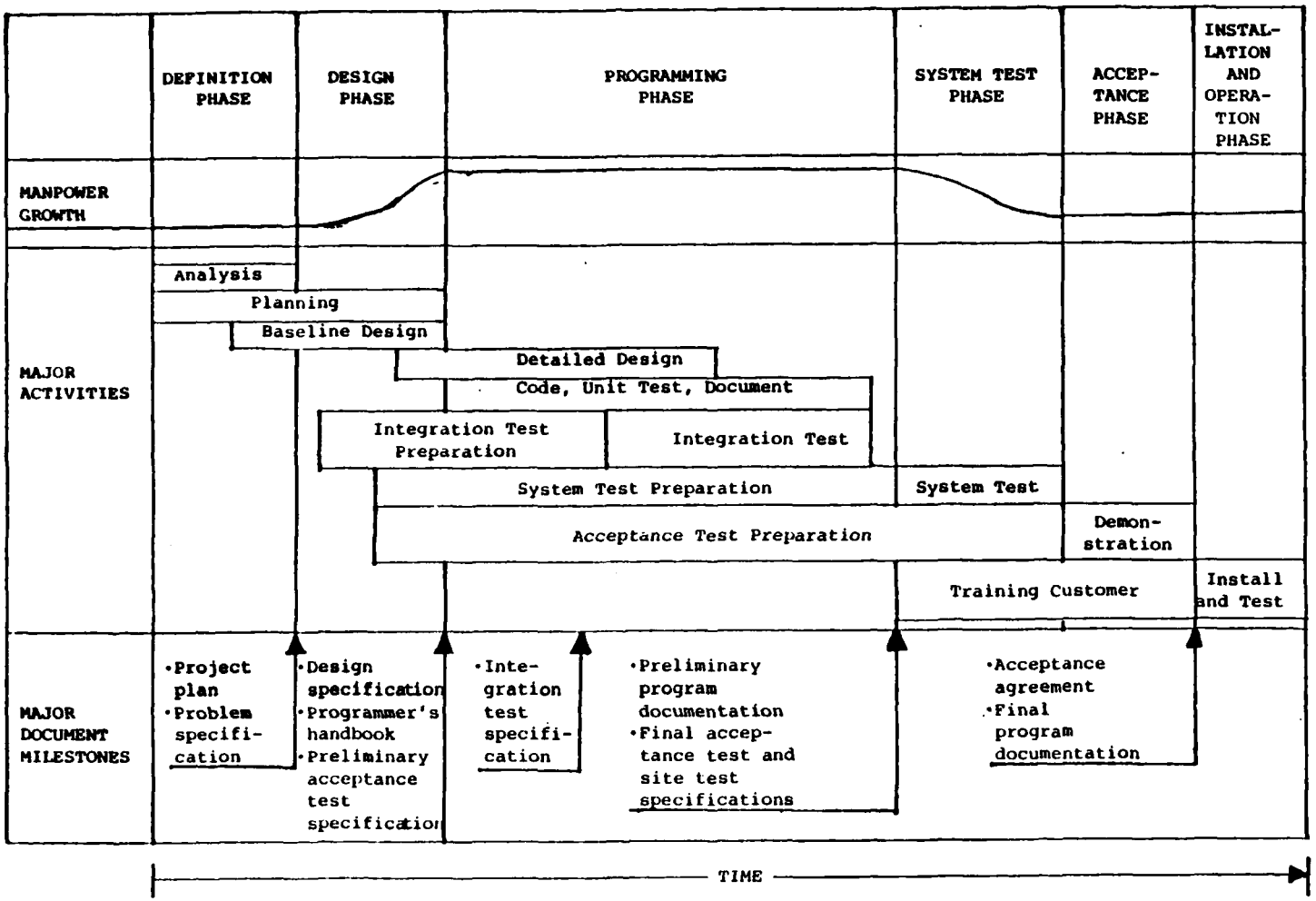


Fig. 3. The program development cycle

done during the planning activity. Teams or committees of people representing the quality, software development, hardware development, and production management areas are utilized to insure that reasonable decisions are made during these phases. It is these first two phases that lay the foundation of the software development project. If a good job is not done at this point, it is extremely difficult to control the development project.

The next major portion of a software development project is the programming phase. It is in this phase that the detailed designing is done and the actual programming takes place. Depending upon the size of the development project, it may be several days or several months after the start of a project before this phase has been reached and any programming is actually done. Each module is programmed, documented, and tested during this phase. Also, the integration of modules with each other and the hardware is tested. Finally, preparation for testing the system under development as an entity is accomplished.

The system test and acceptance phases are the last major phases of a software development project. The installation and operation phase consist of using the software which has been developed and therefore, is not a significant part of the development effort. It is during the system test and acceptance phases that the system is tested as a

whole, customer training is accomplished if needed, and the final documentation is completed.

This brief discussion of the control process suggested by Mr. Metzger was presented so the reader may more fully understand the responses to the research questions. Also, it was Mr. Metzger's text that provided the foundation of NCR's control procedures. In the next sections, each of the research questions and responses will be discussed.

Research Question #1

Does NCR control software development and/or acquisition as a separate element?

Software development and acquisition are controlled as separate items. The new procedure mentioned earlier was implemented for the purpose of controlling software development only. Separately monitored milestones are set for each software development project. Also, cost targets and estimates for software development are maintained, and these targets and estimates are monitored as separate elements. This procedure gives the software a high level of visibility within the management at NCR. Since the separation of control is maintained at all levels of the software development, each manager is able to monitor his progress.

One of the first steps in the control process is the writing of a functional specification. This

specification then becomes the basis for all actions concerning the project for which it was written. During the writing of the specification, decisions are made about developing a certain piece of software or buying it from another source. Milestones are set and cost estimates are also outlined in the specification. Also, the specification provides the foundation from which the quality tests are developed. Each piece of software is tested by quality assurance personnel regardless of whether it was developed by NCR or purchased from another software developer.

Once a project has been started, the changes proposed are also controlled separately by a change control board which consists of people from the quality, product management, software development, and hardware development areas. The changes may come from customers, developers, or any one of many other places. The change control board reviews all proposed changes and makes decisions concerning whether or not they should be implemented on a basis of cost and system requirements. If a change to the project is implemented, the board insures that the proper adjustments are made to the scheduled milestones and the budget forecast of the project.

Modular design and development policies aid the control of software development. The complexity of each module is evaluated by the managers of the project. If it is determined to be too complex, the module is then broken

down into smaller modules or the plans for quality testing are enhanced to insure the module works properly. This policy also makes it easier to implement changes. Modules affected by proposed changes are identified by the change control board in conjunction with the developers and the milestone schedules and cost estimates can be modified to examine the impact of a change on the overall project. Also, minor milestones are set within the major milestones outlined in the functional specification.

Research Question #2

Does NCR involve users throughout the development and/or acquisition process?

Users of software sold by NCR are not involved throughout the process of development or acquisition. Since customer request is one source of new requirements, when a customer requested project is selected by production management, the customer is involved in writing the functional specification. This is accomplished by NCR's marketing personnel who work with the customer to insure the functional specification includes the elements which they wish to purchase. Managers at NCR have found that by not allowing the user to be involved after the specification has been written, more management control can be exercised over the development effort. Customers and future users are allowed to submit changes. However, the changes submitted

by them must go through the change control board for review. This allows the managers of the project to control and coordinate any needed changes.

There are exceptions to this procedure however. In special cases, NCR developers will work with customers. When this is done, special contractual arrangements must be made between NCR and the customer. The special arrangements are negotiated on a case by case basis. This action is required since the more the user is involved, the less management control NCR perceives it has.

Some specific projects may also become an exception due to unique quality test requirements. If quality tests on a particular system are determined by quality assurance managers to be too expensive, then customer test sites are used. In this case, a few customers are selected to use the new product and report the problems encountered. In these cases, users are involved not only with the functional specification, but also during the testing phase.

In every case, after a new product is released, within ninety days to one year after delivery, customers are served by quality assurance personnel. Depending upon the situation, personal interviews, mail, and telephone survey techniques are employed. The results of these surveys are then used by NCR to determine new requirements, areas which need improvements, and areas which contain errors which must be corrected.

Managers at NCR feel that keeping the user from becoming too deeply involved in the development process has improved their products. In support of this, the NCR managers pointed out that most of their large customers preferred to work within NCR's standard procedure and allow NCR to have control of the development effort.

Research Question #3

Does NCR closely integrate the acquisition and/or development of software with hardware?

The managers at NCR are very careful to integrate software with hardware. The integration is considered while the functional specification is being written and also when quality tests are designed, to insure compatability throughout the system under development. Decisions concerning trade-offs between hardware and software are made at the corporate level and incorporated in the functional specification. Also, the functional specification identifies which adjustment or organization will be responsible for each element of the project. In addition, the functional specification identifies a focal point for the project to coordinate each of the separate development efforts.

NCR also makes use of matrix committees and teams made up of people from quality, software, hardware, and other areas. To further aid the integration process, specific attention is paid to the documentation by the

people assigned to the committees and teams. Also, due to this procedure, current knowledge in each area is possessed by the teams and committees.

A great deal of emphasis is placed on the quality test of the interfaces. After a module has been tested by itself, it is then tested to insure proper integration with the hardware. Also, integration of modules with each other is tested by the quality assurance organization. And finally, the hardware and all levels of software are tested together to insure the system is complete and will function properly.

Research Question #4

Does NCR use milestones in the development and/or acquisition process, and if so, how?

NCR does use milestones in the development and acquisition of software. Milestones are key elements in the NCR control process. The major milestones are established in the functional specification. Personnel from quality assurance, production management, and other organizations all work together to set reasonable milestones which can be monitored. Also, cost estimates and targets are set in the functional specification. The functional specification, when completed, becomes a contract between production management and the software developers. If software is purchased outside NCR, the same process is used and the

functional specification is then given to the contractor. Also, dollar limits are set for the purchase of software. If the first limit is exceeded, then the purchase must be reviewed and approved by upper level managers. If the next limit is exceeded, the review and approval is required at the highest level of corporate management.

After the major milestones are given to software development personnel, they are broken down into minor milestones by the developers to insure that the major milestones will be met. Also, if problems do develop, they can be identified and corrected early by monitoring the smaller, more detailed milestones. In addition to breaking down the milestones, Program Evaluation Review Technique (PERT) and Gantt charts are used as an aid to monitor progress with respect to the schedule.

Formal reviews are conducted monthly. Informal reviews are conducted semimonthly or weekly, depending on whether or not the manager involved wants the review. Also, design reviews are conducted when needed. It is during the design reviews that decisions are made concerning whether or not to discontinue a project. When a change has been approved by the change control board, these reviews are used to update the milestones and cost estimates. By conducting the reviews and using the milestones, managers at NCR are aware of the progress being made on each project.

Research Question #5

What effect do the policies identified by the questions above have on the performance, cost, and schedule of software at NCR?

Each of the managers interviewed at NCR feel there has been an improvement since the implementation of their new control procedure. The new procedure has provided more structure to their controls. It has also improved the quality of the functional specifications in their opinion. By providing more structure, the new procedure keeps people from the various organizations working together. Another benefit the managers identified was the ability to measure progress better than before. Also, the current procedure forces the schedule to be met.

A lot of attention is given to the planning of a project. When the milestones and the cost estimates are being made, production management personnel review them to insure they are realistic. Milestones should not be set too close or too far away and cost estimates should not be too low or too high. Under the new procedure, managers are able to meet the milestones and cost targets which have been set. However, since production management reviews the milestones and cost targets, it is not believed that the improvement is due to lengthening milestones and raising cost targets. The new procedure must have made a significant contribution to the improvement.

Benefits in the quality assurance area were also discussed in the interviews. The primary quality measurement discussed was the number of errors found per line of code in the computer program. This indicator is constantly monitored and tracked by the quality assurance organization at NCR and provides a quantitative measurement of the quality of software released. The managers interviewed perceive an improvement in the quality of software purchased from vendors outside NCR. Since software purchased outside must go through the same quality test as software developed by NCR, it is believed that the new procedure has had some impact upon this improvement. Also, some managers felt that a downward trend was developing in the number of errors per line. However, managers also felt that the new procedure had not been used long enough to make any definite statement. In their opinion, it was still too early to tell.

Summary

It was clear to the authors that each of the managers at NCR followed the new procedure. Also, each manager felt that it was an improvement over the methods of control which had been used in the past and had improved their products. Further, since the procedure is flexible, the managers are able to adapt it to individual situations.

The main points brought out as benefits from their procedure were improved functional specifications, focus

on quality assurance, and emphasis on documentation. Since the functional specification provides the foundation for all following actions, its improvement is a benefit. Also, due to the emphasis placed on quality, the managers believe this procedure will aid them in meeting their quality goals. Finally, since NCR managers perceive documentation as one of their primary products, the focus in this area is believed to be beneficial.

In general, the managers prefer the current method of software development control to others that were used in the past. First, it provides them with a structure that allows progress to be measured. Second, it raises the level of visibility of software development. Third, control mechanisms and responsibilities are clearly stated. For each of these reasons, as well as others, managers at NCR feel that they have a good system by which to control and monitor software development.

CHAPTER IV

CONCLUSIONS AND RECOMMENDATIONS

Overview

This chapter represents the "fruit" of the research effort. The foundation of research developed in the preceding chapters has been synthesized into a body of information suitable for inference and action. Structurally, the chapter is divided into conclusions, recommendations, areas of further research, and corollary findings. The conclusions and corollary findings reflect information derived directly from the interviews conducted at NCR by the authors. The recommendations were developed by applying findings to appropriate problem areas in the Air Force. Areas of further research represent avenues of investigation which, in the perception of the authors, present opportunities for greater enhancement of software control.

Conclusions

The method employed by NCR to control software development appears to work well for them. Personnel from the quality, software development, hardware development, and production management areas work closely to establish realistic milestones and cost targets which can be met. This makes the management of a project easier since the

managers can depend upon the milestones and targets being met more consistently. Also, since the NCR managers feel that their current procedure is much better than the previously used procedure, and since the current procedure was designed around software development, as opposed to hardware development, perhaps control procedures for software are not effective when they are modifications of hardware oriented procedures.

Software development is given a high level of visibility in NCR's software control process. First, software management positions exist at nearly all levels of management. Also, the policies and procedures for software development control are established at corporate levels. The high level of visibility is further enhanced since the control procedures for software are separately implemented and monitored. Finally, the use of change control boards aids the coordination of software development with other areas. Since software development has a high level of visibility, problems that develop can be identified quickly and resolved.

A high degree of management control was evident in the software control procedures. The milestones which are set for the software development are used by the managers as tools to aid them in controlling the development process. Similarly, the cost targets are also used to control development. Also, the control procedures call for reviews to be

conducted by the managers of development projects. These reviews aid in keeping the managers informed and knowledgeable about the progress of each project and also serve as a control mechanism. Further, extensive management controls over quality serve to protect the quality and reliability of the end product of the software development effort.

The importance of the involvement of the future users of software throughout the development process may not necessarily be essential. The control procedures employed by NCR do not include the users past the definition and specification phases. Since the new procedures have been implemented, the managers feel that they have been better able to develop software within the milestones and cost targets set for each project. Therefore, the requirement to extensively involve the future users of the software which is being developed may not be a key element to successful projects.

For the reasons stated above, it is the observation of these authors that separate controls over software development are beneficial and help insure that milestones and cost targets are met. Also, a high level of visibility should be given to software development. This is required in order to have separate control of software and to insure management attention is given to software. Further, the use of milestones and cost targets, when set based on the

unique nature of software, aid managers in the identification of problem areas and early solutions. Finally, the importance of user involvement is in doubt. Possibly, when software development is required, the extent and purpose of the project should be examined to determine whether or not the user should be involved during development.

In this section the conclusions which resulted from the findings in Chapter III have been discussed. Software control systems and the basis upon which it was designed was the first item discussed, and second was the visibility of software development. Also, the management control of software development and its relationship to software control was presented. Finally, the necessity to involve future users throughout the development of software was examined. In the next sections, the recommendations and areas for additional research which were extracted from these conclusions will be presented.

Recommendations

Software should become an element in the Work Breakdown Structure. The control of software development would become a separate item and the visibility of software, as well as its control procedures, would be enhanced. Also, with increased visibility on software control, problems in this area could be identified more readily and corrected. Further, by raising software to a visible level of the Work

Breakdown Structure, separate cost estimates and cost targets could be developed. These cost estimates and targets could then be monitored by personnel in the system program office or the contracting officer, whichever is appropriate depending upon the type of contract and/or purchase. In a manner similar to the cost estimates and cost targets, milestones could be set and monitored for the control of software development.

The representative from the using command should know, in detail, what function the software being purchased is to perform. This knowledge is needed in order to write clear and useful software specifications. Also, personnel writing the specifications for the software functions should be very familiar with software. With knowledge in this area, the writer of software specifications is able to include the detail required by the contractors. By having people involved with the software contracts that are knowledgeable in the area of software and what the software being purchased is to do, an accurate assessment of the level of quality which must be achieved, as well as the amount of quality testing which should be done, can be made. The assessment could be made based on the criticality of the function which the software is to perform and the nature of the software, i.e., is the software being developed in a

standard language, has this type of function been programmed before, and are "off the shelf" software packages available.

Software should not be treated as data. If data is to be considered a by-product of a task, and the task being contracted is the development of a software package and its documentation in the form of user manuals and functional descriptions, then the end-product is the software and its documentation. When software is treated as data, the software and its documentation become separate data elements in the contract and the importance of the software documentation becomes less visible when the contracts are reviewed.

Control procedures should be written for the separate control of software development which consider the unique nature of software and are not patterned after hardware control procedures. This, too, would increase the level of visibility of software development control. Also, policies concerning software should consider its unique nature. This is illustrated by the fact that the managers at NCR Corporation discarded their old procedures and developed new ones that did take into co· ideration the unique nature of software as an item, not data. In addition, by having separate control procedures for software development which account for software's uniqueness, the management control of development would be increased. The increase in management's ability to control the software

development would be due to the increased visibility and consideration for the nature of software.

For the reasons stated above, software should be raised to a visible level in the Work Breakdown Structure to provide separate milestones and cost targets for software development. Also, personnel knowledgeable in the area of software and knowledgeable about the function which the software is to perform should be involved in the contracting process to insure that the software's uniqueness and function are both considered in contracts when they are written. Finally, separate control procedures which do not treat software as data, but do consider the nature of software should be written.

Recommendations for Additional Research

In this section, three major areas for further research are discussed. These areas are: (1) quality assurance, (2) Air Force involvement during development, and (3) Software Support. During the course of this research effort, questions in these areas repeatedly surfaced. Therefore, research should be done to clarify what type of relationships exist between software, its development, and the three areas mentioned above.

Quality Assurance

The area of quality assurance as it relates to software, is one in which NCR places a great deal of

emphasis. However, the authors were able to find very little in the Air Force regulations, procedures, and guidebooks which dealt with this area. Therefore, in order to expand the level of knowledge about quality and its relationship to software, further research should be performed.

One way in which research in this area could be conducted is by interviewing or sending questionnaires to personnel in quality assurance departments at other companies in the computer industry. The interviews or questionnaires should be designed to determine how quality assurance in software is managed. From this, the Air Force would be able to evaluate the different methods and develop quality procedures for the software which the Air Force purchases. Also, the importance of integrating quality tests during development could be determined. The question of whether quality tests must be integrated, done as a separate element, or both, could be answered from this type of research. By comparing the level of importance the different companies place on quality in software, the Air Force could better evaluate control procedures and determine how much emphasis should be placed on software quality. Another item of importance is the relationship of software and its documentation. Again, by comparing the policies of companies in this area, the Air Force would be able to understand the relationship and make changes to existing

Air Force policy when required. Finally, through research of this type, the Air Force could develop quality criteria and standards for software.

Another area of qua / assurance in which further research should be conducted is that of automated test equipment. While this is a relatively new area, especially in software, there is knowledge to be gained by research in this area. This research should concentrate on what is currently being done in the area of test equipment for software. Also, some of the procedures, functions, and algorithms pertaining to software test equipment could be uncovered. In addition, the skill levels of personnel which are using, as well as developing, test equipment could be determined. Through the process of gathering and evaluating the information from research in this area, the Air Force could be better able to make decisions concerning the quality of software which it buys and how the quality of software should be tested.

Air Force Involvement

The Air Force has many levels of software requirements ranging from simple accounting and payroll applications to very complex guidance and tracking systems. Since some doubt about the value of having users involved throughout the software development process has been raised,

research should be done as to how much involvement is required at the various levels of complexity. Perhaps software dealing with accounting and payroll type applications could be purchased "off the shelf" through the commercial market. However, for reasons of complexity and secrecy, software for guidance and tracking systems would require a high degree of involvement between Air Force personnel and the contractor. For these reasons, research into the amount of involvement and the benefit received from it at various levels of involvement should be done.

Software Support

Further research is also needed in the area of follow-on support to software once it has been developed and purchased. Due to modifications in weapon systems, mission requirements, and other equipment the software incorporated into these pieces of equipment must also be modified. Also, as the state of the art advances, new and better methods of utilizing software are being developed which cause current systems to become outdated. For these reasons, exploratory research into how to handle software support is recommended. Perhaps this could be done best by asking companies that currently develop and support software what their policies and procedures are. Regardless of how the research is done, more knowledge should be obtained

by the Air Force in this area because as more software is purchased the problem of software support will grow.

Corollary Findings

In the course of conducting the research, a number of findings or observations, not directly related to the research effort, were made. Although the aforementioned findings did not contribute significantly in answering the research questions, they provide additional insight into software control and general software management.

The first corollary finding reinforces an observation made by Johns Hopkins University in a management study: hardware oriented control philosophies and procedures don't work well when applied to software (10:2-4). An integral part of NCR's software control system was the selective abandonment of software management policies adapted directly from hardware management procedures. NCR realized that "crossing out the word hardware and entering software" in written guidance does not necessarily provide the needed procedures for managing software. The recognition of this fact was perceived by NCR software managers as an essential precursor to an effective software control system.

The second corollary finding of signifiance was the great importance and contribution of Quality Assurance personnel to the software control program. Operating autonomously and semiautonomously within specific departments,

Quality Assurance functionaries rigorously test software operation, hardware and software integration, and validate documentation. The tests are made at specific and well defined phases or milestones using defined standards, statistical techniques, and state of the art test equipment. Quality discrepancies can be sequentially elevated to the highest levels of management for problem resolution. Corporate Quality Assurance is the focal point for serious discrepancies and they have direct access to the highest decision making authority. NCR software managers perceive that this tremendous emphasis on quality has markedly reduced errors in software products.

The third and final corollary finding deals with the application of milestones and cost controls as a mechanism in managing software. In the main body of research, the presence of milestones and cost controls were primarily viewed as a barometer of software visibility. During the course of the research, the perceived usefulness of milestones and cost controls in software control was revealed. Milestones and cost controls, tailored to the unique nature of software, were used as effective tools in meeting and measuring schedule, quality, and fiscal standards. NCR managers perceive that modular design and development was particularly well suited to the use of milestones and cost controls.

Summary

This fourth and final chapter is the culmination of the research effort. The authors have endeavored to present inferences and recommendations warranted by the nature and depth of the research. The corollary findings and conclusions presented reflect the perceptions of seven experienced software managers at NCR. The recommendations are derived from a synthesis of findings, areas of perceived Air Force software deficiency, and a need for amplification or exploration of additional aspects of software control. This research purports not to provide the entire solution to the software control problem, but to provide a contribution to the ultimate solution. The authors believe that research enacted upon previously stated recommendations will add more pieces to the software control puzzle.

As stated in the first chapter, software is the essential and most costly aspect of computerized systems. The proliferation of software using systems is increasing every year. It is incumbent upon the Air Force to minimize the cost and maximize the effectiveness of software acquisitions. A well conceived software control policy is essential to meet that goal. The authors believe that this research provides a foundation upon which to build an effective software control policy.

SELECTED BIBLIOGRAPHY

A. REFERENCES CITED

1. Boehm, B. W. "Software and Its Impact: A Quantitative Assessment," Datamation, May 1973.

2. Carroll, Archie B., and Hugh J. Watson. Computers for Business. Dallas: Business Publications, Inc., 1976.

3. Davis, M. R., R. N. Reinstedt, and R. Turn. "A Management Approach to the Development of Computer-Based Systems." Rand Report P-5686, Santa Monica CA, July 1976.

4. DeRoze, Barry C., and Thomas H. Nyman. "The Software Life Cycle--A Management and Technological Challenge in the Department of Defense," IEEE Transactions on Software Engineering, July 1978, pp. 309-318.

5. Drezner, S. M., and others. "The Computer Resources Management Study." Rand Report R-1855/1PR, Santa Monica CA, April 1976.

6. Driscoll, Lieutenant Colonel Alan J., USAF. "Software Visibility and the Program Manager," Defense Systems Management Review, Spring 1977.

7. Electronic Systems Division, Air Force Systems Command. A Review of Software Cost Estimation Methods. ESD-TR-76-271. Washington: Government Printing Office, August 1976.

8. Emory, William C. Business Research Methods. Homewood IL: Richard D. Irwin, Inc., 1976.

9. Fisher, David A. "DOD's Common Programming Language Effort," IEEE Transactions on Software Engineering, March 1978, pp. 24-33.

10. Johns Hopkins University Applied Physics Laboratory. DOD Weapon Systems Software Management Study. Washington: Office of Assistant Secretary of Defense (I&L). AD-ADZZ-160, June 1975.

11. Marciniak, Lieutenant Colonel John H., USAF. "Software Acquisition Within Air Force System Command--A Management Approach," Defense Systems Management Review. Washington: Government Printing Office, 1978, pp. 32-39.

12. Metzger, Philip W. Managing a Programming Project. Englewood Cliffs NJ: Prentice Hall, Inc., 1973.

13. Rome Air Development Center. Software Data Collection Study. Griffiss Air Force Base NY, December 1976.

14. Royce, Winston W. "Software Requirements Analysis, Sizing Plus Costing." Proceedings, TRW Symposium on Reliable, Cost-Effective, Secure Software, March 1974.

15. Slay, General Alton D., USAF, Commander Air Force Systems Command. Address to NCMA 18th Annual National Symposium, Los Angeles, July 26-27, 1979.

16. Value Line Investment Survey: Ratings and Reports. New York: Arnold Bernhard and Co., Inc., November 16, 1979.

B. RELATED SOURCES

Aeronautical Systems Division, Air Force Systems Command. Management Guide to Avionics Software Acquisition. ASD-TR-76-11. Washington: Government Printing Office, June 1976.

_____. Management Guide to Avionics Software Acquisition: An Overview of Software Development and Management. ASD-TR-76-11. Washington: Government Printing Office, June 1976.

_____. Management Guide to Avionics Software Acquisition: Summary of Software Related Standards and Regulations. ASD-TR-76-11. Washington: Government Printing Office, June 1976.

_____. Management Guide to Avionics Software Acquisition: Technical Aspects Relative to Software Acquisition. ASD-TR-76-11. Washington: Government Printing Office, June 1976.

Electronic Systems Division, Air Force Systems Command. An Air Force Guide to the Computer Program Development Specification. ESD-TR-78-139. Washington: Government Printing Office, November 1977.

_____. Cost Reporting Elements and Activity Cost Trade-Offs for Defense System Software (Executive Summary). ESD-TR-78-139. Washington: Government Printing Office, May 1977.

_____. Cost Reporting Elements and Activity Cost Trade-Offs for Defense System Software (Study Results). ESD-TR-77-262. Washington: Government Printing Office, May 1977.

_____. Life Cycle Cost/Design-to-Cost Guidelines. ESD-TR-75-77. Washington: Government Printing Office, June 1975.

_____. Software Acquisition Management Guidebook: Cost Estimation and Measurement. ESD-TR-78-140. Washington: Government Printing Office, March 1978.

_____. Software Acquisition Management Guidebook: Life Cycle Events. ESD-TR-77-22. Washington: Government Printing Office, February 1977.

_____. Software Acquisition Management Guidebook: Series Overview. ESD-TR-78-141. Washington: Government Printing Office, March 1978.

_____. Software Acquisition Management Guidebook: Software Maintenance. ESD-TR-77-327. Washington: Government Printing Office, October 1977.

Office of the Secretary of Defense. Embedded Computer Resources and the DSARC Process. Washington: Government Printing Office, 1977.

Rome Air Development Center. Software Cost Estimation Study. Guidelines for Improved Software Cost Estimation. RADC-TR-77-220. Washington: Government Printing Office, August 1977.

_____. Software Data Collection Study. Data Requirements for Productivity and Reliability Studies. RADC-TR-76-329. Washington: Government Printing Office, December 1976.

U.S. Department of Defense. Acquisition and Support Procedures for Computer Resources in Systems. AFR-800-14. Washington: Government Printing Office, 1975.

_____. Management of Computer Resources in Systems. AFR-800-14. Washington: Government Printing Office, 1975.